

```
Ввод [5]: import numpy as np
import pandas as pd
import scipy.stats as st
import matplotlib.pyplot as plt

from statsmodels.stats.weightstats import _zconfint_generic, _tconfint_generic
from statsmodels.stats.proportion import proportion_confint
```

```
Ввод [2]: data = pd.read_csv('data.csv')
```

```
Ввод [3]: data.head()
```

```
Out[3]:
```

	id	required_amt	installment_flg	monthly_income_amt	age	has_high_education_flg	approved_flg
0	1	23906.0	0	80000	37	0	0
1	2	13111.0	1	43000	22	0	0
2	3	43266.0	1	20000	34	0	0
3	4	68782.0	1	35000	24	0	0
4	5	19550.0	1	25000	42	1	0

- id - идентификатор заявки на получение кредита на товар/услугу
- required_amt - запрошенная сумма кредита
- installment_flg - флаг промо продукта
- monthly_income_amt - ежемесячный доход заявителя
- age - возраст заявителя
- has_high_education_flg - наличие высшего образования
- approved_flg - одобрил ли банк заявку

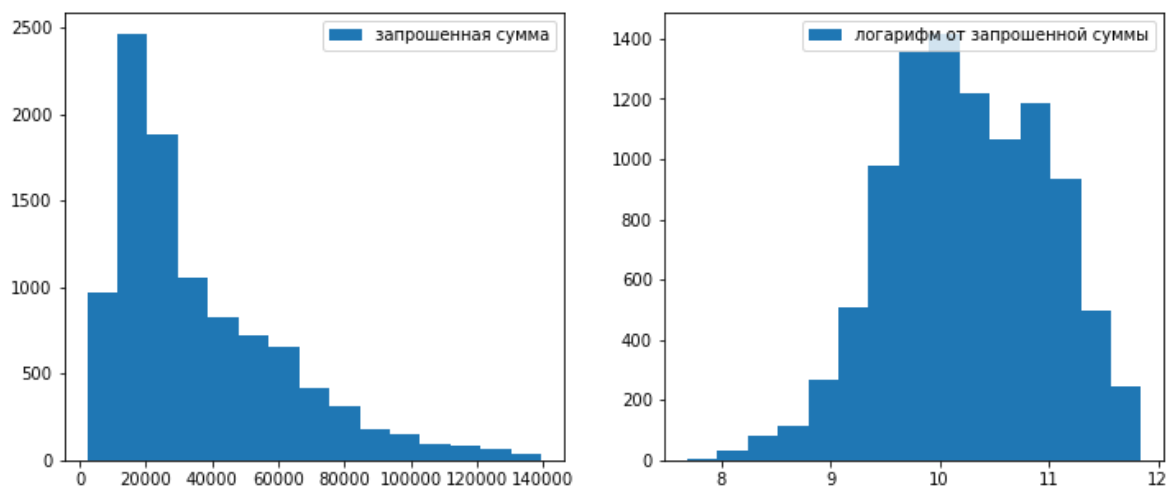
```
Ввод [46]: fig,ax = plt.subplots(1,2)

fig.set_size_inches(12,5)
ax[0].hist(data.required_amt[data.required_amt < data.required_amt.quantile(0.99)],
            bins=15, label = 'запрошенная сумма')
ax[0].legend()

ax[1].hist(np.log(data.required_amt[data.required_amt < data.required_amt.quantile(0.99)]),
            bins=15, label = 'логарифм от запрошенной суммы')
ax[1].legend()

plt.plot()
```

```
Out[46]: []
```



Построить предсказательные интервалы для ежемесячного дохода и запрошенной суммы (в предположении

логнормальности данных)

$$\log X \sim N(\mu, \sigma^2)$$

Тогда

$$\frac{\log X - \mu}{S} \sim t_{n-1}$$

В таком случае предсказательный интервал примет вид

$$\mathbf{P} \left\{ \exp\{\hat{\mu} - t_{n-1, 1-\frac{\alpha}{2}} S\} < X < \exp\{\hat{\mu} + t_{n-1, 1-\frac{\alpha}{2}} S\} \right\} \approx 1 - \alpha$$

```
Ввод [67]: sample_req = data[ (data.required_amt < data.required_amt.quantile(0.99)) & (data.required_amt >
sample_inc = data[ (data.monthly_income_amt < data.monthly_income_amt.quantile(0.99)) & (data.mo
```

```
Ввод [68]: log_req_pred_int= _zconfint_generic(np.log(sample_req).mean(),
                                              np.var(np.log(sample_req)),
                                              0.05,
                                              'two-sided'
                                              )
log_inc_pred_int= _zconfint_generic(np.log(sample_inc).mean(),
                                    np.var(np.log(sample_inc)),
                                    0.05,
                                    'two-sided'
                                    )

print(f'required_amt 95% predictive interval: [{np.exp(log_req_pred_int[0]):,.2f}] - {np.exp(log_r
print(f'monthly_income_amt 95% predictive interval: [{np.exp(log_inc_pred_int[0]):,.2f}] - {np.exp

required_amt 95% predictive interval: [10,947.75] - 76,428.13 ]
monthly_income_amt 95% predictive interval: [25,161.05] - 62,566.92 ]
```

Ввод []:

Ввод []: